COSC 3360/6310 FIRST QUIZ ANSWERS

Jehan-François Pâris jfparis@uh.edu

Summer 2017

What is the difference between *interactive* applications and *real-time applications*?

What is the difference between *interactive* applications and *real-time applications*?

□Interactive applications do not have strict deadlines.

Why is *memory protection* always implemented in *hardware?*

Why is *memory protection* always implemented in *hardware?*

□ Because it needs to be done very fast.

Why is it impossible to hijack interrupts?

Why is it impossible to hijack *interrupts?*

□ Because their destination addresses are hard-wired.

What is the purpose of the argv parameter in the execv() system call?

What is the purpose of the argv parameter in the execv() system call?

It points to a list of parameters to be passed to the program fetched by the execv() system call.

• Why are most servers *multithreaded?*

• Why are most servers *multithreaded?*

□ In order to be able to handle multiple requests in parallel.

Give an example of a *soft real-time* application.

Give an example of a *soft real-time* application.

□ Watching a video.

Main advantage of modular kernels:

Main advantage of modular kernels:

They let users add functionality to the kernel without having to recompile it (and without having to endure additional delays).

Main disadvantage of delayed writes:

Main disadvantage of delayed writes:

□ They do not guarantee the durability of writes.

Main advantage of monolithic kernels:

Main advantage of monolithic kernels:

□ They are fast.

Main advantage of symmetric organization for multiprocessor OSes:

Main advantage of symmetric organization for multiprocessor OSes:

They avoid potential bottlenecks (by letting the kernel to run on each processor).

Main advantage of lightweight processes over conventional processes:

Main advantage of lightweight processes over conventional processes:

They can be created much more cheaply than conventional processes because they share the address space of their parent.

Main disadvantage of hard drives:

Main disadvantage of hard drives

□ They are by far the slowest component of any computer system.

Third question

```
Mhat will the following program print out?
    [main() {
        fork();
        if (fork() == 0) {
            cout << "Hello!\n";
        }
        cout << "Goodbye!\n";
    }
}</pre>
```

What will happen (II)



What will happen (II)



Program will print two times "Hello!" and four time "Goodbye!"

Third question

The program will print out <u>two</u> "Hello!" and <u>four</u> "Goodbye!"

Complete the following sentences:

We can safely swap out processes that have remained a long time in the state.

When a process does a blocking system call, it remains in the ______ state until the call is completed then goes to the state.

Complete the following sentences:

- We can safely swap out processes that have remained a long time in the <u>blocked</u> state.
- When a process does a blocking system call, it remains in the ______ state until the call is completed then goes to the state.

Complete the following sentences:

- We can safely swap out processes that have remained a long time in the <u>blocked</u> state.
- When a process does a blocking system call, it remains in the <u>blocked</u> state until the call is completed then goes to the state.

Complete the following sentences:

- We can safely swap out processes that have remained a long time in the <u>blocked</u> state.
- □ When a process does a blocking system call, it remains in the <u>blocked</u> state until the call is completed then goes to the <u>ready</u> state.

- What is the default action a Linux process takes when it receives a *signal*?
- How can a process specify a *different action*?
- Is it always possible?

What is the default action a Linux process takes when it receives a *signal*?

□ It terminates.

- How can a process specify a *different action*?
- Is it always possible?

What is the default action a Linux process takes when it receives a *signal*?

□ It terminates.

- How can a process specify a *different action*?
 It can catch the signal using signal().
- Is it always possible?

What is the default action a Linux process takes when it receives a *signal*?

□ It terminates.

- How can a process specify a *different action*?
 It can catch the signal using signal()
- Is it always possible?

□No, the SIGKIL signal cannot be caught.